ABSTRACT

Testing is a critical activity since a single fault can cripple a whole system and result in great loss. Bugs in computer hardware and software are no more than the crystallization in silicon and plastic of the mental mistakes all people make. People are only human, after all so computers can only reflect our own humanity.

Testing is expensive. It often consumes between 1/3 to 1/2 of the total cost of software development. Let us remember the Year 2000 problem. The total cost world-wide of changing and testing the relevant software was estimated to be 400 billion dollars [ECON 97].

Testers face the challenge of doing testing within the constrained schedule. In the hyper competitive commercial market place, it is not practical to exhaustively test all combinations of system test cases. The number of ways a system must be tested can often be overwhelming. There are a number of automatic test case generation tools available but these can suffer from combinatorial explosions in the number of possibilities to test.

The combinatorial approach to software testing uses models to generate an optimal number of test inputs so that selected combinations of input values are covered. The most common coverage criteria is two-way, or pair-wise coverage of value combinations, though for higher confidence three-way or higher coverage may be required. This system presents examples of requirements and related models for applying the combinatorial approach to those requirements. The paradigm of model-
based testing shifts the focus of testing from writing individual test cases to developing a model from which a test suite can be generated automatically.

Given the complexity of testing, the model based testing approach is used in conjunction with test automation harnesses. Since no large empirical study has been conducted to measure efficacy of this new approach, we report on our experience with developing tools and methods in support of model-based testing.

The RATG System (Reduced Automated Test case Generator), developed as a part of our research work, discusses the method that generates an optimal subset of test cases, which provides good coverage of the test domain using Combinatorial Design Method. This approach uses combinatorial designs to generate tests that cover the pair-wise, triple or n-way combinations of a system's test parameters. Although this system focuses on pair-wise coverage, the discussion is equally valid when higher coverage criteria such as three-way (triples) are used. These parameters determine the system's test scenarios. This new method is implemented in the RATG system. The RATG system can be used in a variety of applications to test for unit, system, and interoperability testing. It can generate both high-level test plans and detailed testcases. In most applications, it can greatly reduce the cost and time schedule of test plan development.